UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/755,564	01/05/2001	Marc Lamberton	FR920000023US1	4934

36736 7590 08/09/2005

DUKE W. YEE
YEE & ASSOCIATES, P.C.
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
TRAN, ELLEN C	
ART UNIT	**PAPER NUMBER**
2134	

DATE MAILED: 08/09/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

ɫ

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents

United States Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

www.uspto.gov

MAILED

AUG 09 2005

Technology Center 2100

BEFORE THE BOARD OF PATENT APPEALS

AND INTERFERENCES

μ

Application Number: 09/755,564

Filing Date: January 05, 2001

Appellant(s): LAMBERTON ET AL.

Theodore D. Fay, III

For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 23 May 2005.

(1) Real Party in Interest

IBM Corporation

(2) Related Appeals and Interferences

None

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments After Final

No After Final Amendments were filed.

(5) Summary of Invention

The summary of invention contained in the brief is correct.

(6) Issues

The appellant's statement of issues in the brief is correct

(7) Claims Appealed

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Prior Art of Record

5,958,053 Denker 28 September 1999

TCP/IP Illustrated Volume 1 The Protocols by W. Richard Stevens 1994, page 231

(9) Grounds of Rejection

The following grounds of rejection are applicable to the appealed claims:

Claims 1-5 and 7-14 are rejected under 35 U.S.C. 102(e). This rejection is set forth in a prior Office Action, mailed on 13 January 2005.

Regarding claim 1, the previously stated rejection is based upon U.S. Patent No. 5,958,053 to Denker (hereinafter '053).

As per the first limitation of claim 1, **"A method for defeating, in a server unit of an Internet Protocol network, a SYN flooding attack"** is taught in '053 col. 4, lines 48-52 "The protocols of the present invention provide excellent protection against SYN Flooding because minimal resources are allocated in response to a SYN message, while permitting the server to implement all client requested options".

As per the second limitation, **"said server unit running Transport Control Protocol to allow the establishment of one or more transmission control protocol connections with one or more client units"** is shown in '053 col. 6, lines 19-64 "Network Environment ... Network environment 100 includes multiple devices including client 105 and server 110 ... Each of Client 105 and server 110 includes an implementation of TCP and IP ... The present invention is applicable to a wide variety of computers and other devices".

As per the third limitation, **"said method comprising the steps of: upon having activated the transmission control protocol in said server unit: listening for the receipt of a SYN message sent from a client unit" and "resuming to said listening step"** is disclosed in '053 col. 6, lines 59-60 "The term "server" means a device which listens for and accepts TCP connection request".

As per the fourth limitation, **"upon receiving said SYN message: computing an Initial Sequence number Receiver side"** is taught in '053 col. 7, lines 5-46 "Server 110 then receives the SYN message of step 1 from client 105, state 20C. After receiving the SYN message of step 1020C, server 110 performs only the minimal communication and computation, and allocates no

memory resources for the incipient connection. Server 110 sends a SYNACK message (in step 2030C) to client 105, with the sequence number set to 300 (in the example), the acknowledgement number set to 101 to acknowledge the SYN message to step 1020C". _The computed initial sequence number Receiver side is 101, see FIG. 4, 2030C._

As per the fifth limitation, **"wherein said Initial Sequence number Receiver side is embedded with connection parameters specified in the SYN message"** is shown in '053 col. 7, lines 46-58 "The SYNACK message (in step 2030C) also includes an encoded value (represented in FIG. 4 as $c) ". _The connection parameters specified in the SYN message are $c which can include additional parameters such as "connection parameters" for example client's port, the server's IP address, the server's port, see FIG. 4 2030C._

As per the sixth limitation, **"responding to said client unit with a SYN-ACK message including said computed said Initial Sequence number Receiver side"** is disclosed in '053 col. 7, lines 32-46 "Server 110 then receives the SYN message of step 1 from client 105, state 20C. After receiving the SYN message of step 1020C, server 110 performs only the minimal communication and computation, and allocates no memory resources for the incipient connection. Server 110 sends a SYNACK message (in step 2030C) to client 105, with the sequence number set to 300 (in the example), the acknowledgement number set to 101 to acknowledge the SYN message to step 1020C".

As per the seventh limitation, **"responsive to receiving an ACK message, determining whether to establish a transmission control block for the client unit by evaluating an incremented value of the Initial Sequence number Receiver side included in the ACK message"** is shown in '053 col. 8, lines 14 through col. 9, line 1 "At state 30C, client 105

receives the SYNACK message of step 2030C and the connection is established on client 105's end ... In step 3040C, client 105 then sends an ACK message to server 110 with a sequence number set to 101, an acknowledgement number set to 301 to acknowledge the SYNACK message of step 2030C, and with the ACK flag set ... After server 110 receives the ACK message of step 3040C, server 110 analyzes the encoded value passed in the ACK message of step 3040C to determine if it passes an appropriate mathematical (i.e., cryptologic) test. As one example of the appropriate mathematical test, server 110 recalculates the output of the mathematical function used previously by server 110 (to calculate the encoded value $c) using the parameters passed in the message of step 3040C". *Note the Initial Sequence number Receiver side of 101 is included in the ACK message see FIG. 4, 3040C, as per TCP.*

The TCP handshake is well known in the art for connection establishment, the reference '053 shows incrementing the sequence number in SYN messaged passed (from 100 to 101 / from 300 to 301) in the example provided. Incrementing the sequence number by one in order to acknowledge the TCP connection in relation to '053 FIG. 4, is a basic operation of TCP; see TCP/IP Illustrated Volume 1 The Protocols by W. Richard Stevens page 231 (copyrighted in 1994).

Claims 2, 11, 12, and 14 stand or fall with claim 1. Claims 11 and 14 are independent claims containing limitation similar to those present in claim 1.

Regarding claim 3, **"wherein said computing step further comprises the steps of: updating, in said server unit, a pseudo-random number (PRN) generator holding a current key; remembering a former key; and using said current key as said randomly generated key for said computed Initial Sequence number Receiver side"** is shown in '053 col. 5, lines

16-25 "If the client's address is not on the server's Friends Table, the server calculates an encoded value. The encoded value is calculated as the mathematical function of at least the client's address and a secret (i.e. random number) known only to the server. The server sends and ACK message to the client including the calculated encoded value as the acknowledgment number".

Claims 10 and 13 stand or fall with claim 3.

Regarding claim 4, **"wherein the step of concatenating said server signature and said category index further includes the step of picking up a category index within said set of predefined connection categories on the basis of the content of said received SYN message"** is disclosed in 053 col. 7, lines 47-67 "The SYNACK message of step 2030C also includes an encoded value (represented in FIG. 4 as $c). For security reasons, the encoded value $c can be calculated by server 110 as a cryptologic function (or mathematical function) that depends upon at least the IP address of client 105 and a secret only known to server 110. The encoded value $c can be a crytologic function which depends upon one or more additional parameters (in addition to the secret and the IP address of client 105), including: the client's port, the server's IP address, the server's port, and the clients' sequence number, and other things. For example the encoded value $c an be calculated by server as follows $c=MD5 has(client's IP address, client's port, server's IP address, server's port, random secret) + client's initial sequence number ".

Regarding claim 5, **"wherein said updating step includes the step of: updating said PRN generator at a rate not higher than an Maximum Segment Lifetime defined in said transmission control protocol connections"** is taught in '053 col. 8, lines 60 through col. 9, line 25 "After server 110 receives the ACK message of step 3040C ... As one example of the

appropriate mathematical test, server 110 recalculates the output of the mathematical function used previously by server 110 (to calculate the encoded value $c) using the parameters passed in the message of step 3040C. In an embodiment where the encoded value $c is calculated as a mathematical function of the client's IP address and the secret known only to server 110, server 110 recalculates the encoded value using the client IP address provided in the message of step 3040C and the secret. In the embodiment where the encoded value $c of step 2030C was calculated as a cryptologic function of equation 1, the new calculated value can be calculated by server 110 as:

New calculated value MD5 hash (client's IP address$_{msg3}$, client's port$_{msg3040C}$,
server's IP address $_{msg3040C}$ server's port$_{msg3040C}$, random secret)+
(client's sequence number$_{msg3040c\ -1}$). (Eq. 2).

Equation 2 states that the new calculated value can be calculated as the MD5 hash function of the random secret and the following parameters contained in the message of step 3040C: (the client's IP address, the client's port, the server's IP address, the server's port), plus the (sequence number in the message of step 3040C)-1".

Regarding claim 7, as per the first limitation of claim 7, **"A method for defeating, in a server unit of an IP network, a SYN flooding attack"** is taught in '053 col. 4, lines 48-52 "The protocols of the present invention provide excellent protection against SYN Flooding because minimal resources are allocated in response to a SYN message, while permitting the server to implement all client requested options".

As per the second limitation, **"said method comprising the steps of listening for an ACK message sent from a client unit"** and **"resuming to said listening step"** is shown in '053 col. 6, lines 59-60 "The term "server" means a device which listens for and accepts TCP connection request".

As per the third limitation, **"upon receiving said ACK message, evaluating a value of an Initial Sequence number Receiver side included that includes content comprising embedded connection parameters specified in a previously received SYN message as an authentic computed Initial Sequence number Receiver side"** is disclosed in '053 col. 8, lines 14 through col. 9, line 1 "At state 30C, client 105 receives the SYNACK message of step 2030C and the connection is established on client 105's end ... In step 3040C, client 105 then sends an ACK message to server 110 with a sequence number set to 101, an acknowledgement number set to 301 to acknowledge the SYNACK message of step 2030C, and with the ACK flag set ... After server 110 receives the ACK message of step 3040C, server 110 analyzes the encoded value passed in the ACK message of step 3040C to determine if it passes an appropriate mathematical (i.e., cryptologic) test. As one example of the appropriate mathematical test, server 110 recalculates the output of the mathematical function used previously by server 110 (to calculate the encoded value $c) using the parameters passed in the message of step 3040C". *Note the Initial Sequence number Receiver side of 101 is included in the ACK message see FIG. 4, 3040C, as per TCP. The connection parameters specified in the SYN message are $c which can include additional parameters such as "connection parameters" for example client's port, the server's IP address, the server's port, see FIG. 4 2030C.*

As to the fourth limitation, "**and responsive to evaluating the value of the Initial Sequence Number Receiver side as an authentic computed Initial Sequence number Receiver side, allocating resources for a transmission control protocol connection according to said content; and**" is taught in '053 col. 9, lines 20-33 "Server 110 then compares the new calculated encoded value to the encoded value $c received by server 110 in the message of step 3040C. If these two values match, then the message of step 3 passes the cryptologic test and client 105 is properly complying ... Server 110 then allocates a full Transmission Control Block in memory for storing all required information regarding the connection with client 105 (including the client requested options), and the TCP connection is now fully established, step 40C".

The TCP handshake is well known in the art for connection establishment, the reference '053 shows incrementing the sequence number in SYN messaged passed (from 100 to 101 / from 300 to 301) in the example provided. Incrementing the sequence number by one in order to acknowledge the TCP connection in relation to '053 FIG. 4, is a basic operation of TCP; see TCP/IP Illustrated Volume 1 The Protocols by W. Richard Stevens page 231 (copyrighted in 1994).

Claims 8 and 9 stand or fall with claim 7.

(10) Response to Arguments

Regarding Appellant's argument 1, "that Denker does not anticipate claim 1 because Denker does not show or suggest the claimed step of: responsive to receiving and ACK message, determining whether to establish a transmission block from the client unit by evaluating an incremented value of the Initial Sequence number Receiver side included in the ACK message".

The grounds of rejection stated above show that the invention disclosed by Denker receiving and evaluating an ACK message received with an increment value of the Initial Sequence number Receiver side. It is noted that the incremented Initial Sequence number is a known element of the standard TCP protocol.

Regarding Appellant's second argument, "Denker does not anticipate claim 3 because Denker does not show the steps of updating a PRN, holding a current key, remembering a current key, and using a current key as claimed".

The grounds of rejection stated above shown that the invention disclosed by Denker updates or recalculates the key. The random number is generated at the server and used in calculation of the encoded value which is used in the messages exchanged when establishing the connection.

Regarding Appellant's third argument, "Denker does not anticipate claim 4 because Denker does not disclose picking a category index within said set of connection categories on the basis of content said SYN message, as claimed".

The grounds of rejection stated above shown that the invention disclosed by Denker uses an encoded value $c which includes various parameters including connection parameters (i.e. server port number, client port number, IP address).

Regarding Appellant's fourth argument, "Denker does not anticipate claim 5 because Denker shows none of the limitation of claim 5 ... the cited passage manifestily does not show or suggest updating a PRN generator at a maximum rate as claimed".

The grounds of rejected stated above show the secret (which can be or include a random number) is recalculated. The rate at which the secret is updated is in accordance with the

transmission protocol as claimed, not at a maximum rate as argued. Claim 5 states: "at a rate not higher than a Maximum Segment Lifetime defined in said transmission control protocol connections".

Regarding Appellant's fifth argument, "Denker does not anticipate claim 7 because Denker does not show the step of: responsive to evaluating the value of the Initial Sequence Number Receiver side".

The ground of rejection stated above show the evaluation of the ISN receiver side. The ISN receiver side in the example provided as well as in FIG. 4 is 101, the evaluation of the message containing 101 is shown in the rejection above.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Ellen Tran
Patent Examiner
Technology Center 2134
1 August 2005

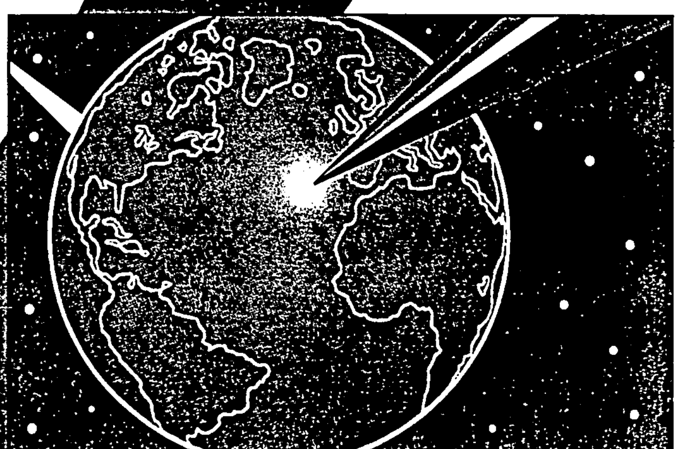
GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Conferees:
Gregory Morse
David Jung
DJ.

TCP/IP Illustrated Volume 1

The Protocols

W. Richard Stevens



ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

The publisher offers discounts on this book when ordered in quantity for special sales.
For more information please contact:
Corporate & Professional Publishing Group
Addison-Wesley Publishing Company
One Jacob Way
Reading, Massachusetts 01867

Library of Congress Cataloging-in-Publication Data
Stevens, W. Richard
TCP/IP Illustrated: the protocols/W. Richard Stevens.
p. cm. — (Addison-Wesley professional computing series)
Includes bibliographical references and index.
ISBN 0-201-63346-9 (v. 1)
1.TCP/IP (Computer network protocol) I. Title. II. Series.
TK5105.55S74 1994
004.6'2—dc20

Text printed on recycled and acid-free paper.

ISBN 0201633469

14 1516171819 MA 02 01 00 99

14th Printing July 1999

In line 1, the field `1415531521:1415531521(0)` means the sequence number of the packet was 1415531521 and the number of data bytes in the segment was 0. `tcpdump` displays this by printing the starting sequence number, a colon, the implied ending sequence number, and the number of data bytes in parentheses. The advantage of displaying both the sequence number and the implied ending sequence number is to see what the implied ending sequence number is, when the number of bytes is greater than 0. This field is output only if (1) the segment contains one or more bytes of data or (2) the SYN, FIN, or RST flag was on. Lines 1, 2, 4, and 6 in Figure 18.1 display this field because of the flag bits—we never exchange any data in this example.

In line 2 the field `ack 1415531522` shows the acknowledgment number. This is printed only if the ACK flag in the header is on.

The field `win 4096` in every line of output shows the window size being advertised by the sender. In these examples, where we are not exchanging any data, the window size never changes from its default of 4096. (We examine TCP's window size in Section 20.4.)

The final field that is output in Figure 18.1, `<mss 1024>` shows the *maximum segment size* (MSS) option specified by the sender. The sender does not want to receive TCP segments larger than this value. This is normally to avoid fragmentation (Section 11.5). We discuss the maximum segment size in Section 18.4, and show the format of the various TCP options in Section 18.10.

Time Line

Figure 18.3 shows the time line for this sequence of packets. (We described some general features of these time lines when we showed the first one in Figure 6.11, p. 80.) This figure shows which end is sending packets. We also expand some of the `tcpdump` output (e.g., printing SYN instead of `S`). In this time line we have also removed the window size values, since they add nothing to the discussion.

Connection Establishment Protocol

Now let's return to the details of the TCP protocol that are shown in Figure 18.3. To establish a TCP connection:

1. The requesting end (normally called the *client*) sends a SYN segment specifying the port number of the *server* that the client wants to connect to, and the client's *initial sequence number* (ISN, 1415531521 in this example). This is segment 1.
2. The server responds with its own SYN segment containing the server's initial sequence number (segment 2). The server also acknowledges the client's SYN by ACKing the client's ISN plus one. A SYN consumes one sequence number.
3. The client must acknowledge this SYN from the server by ACKing the server's ISN plus one (segment 3).

These three segments complete the connection establishment. This is often called the *three-way handshake*.